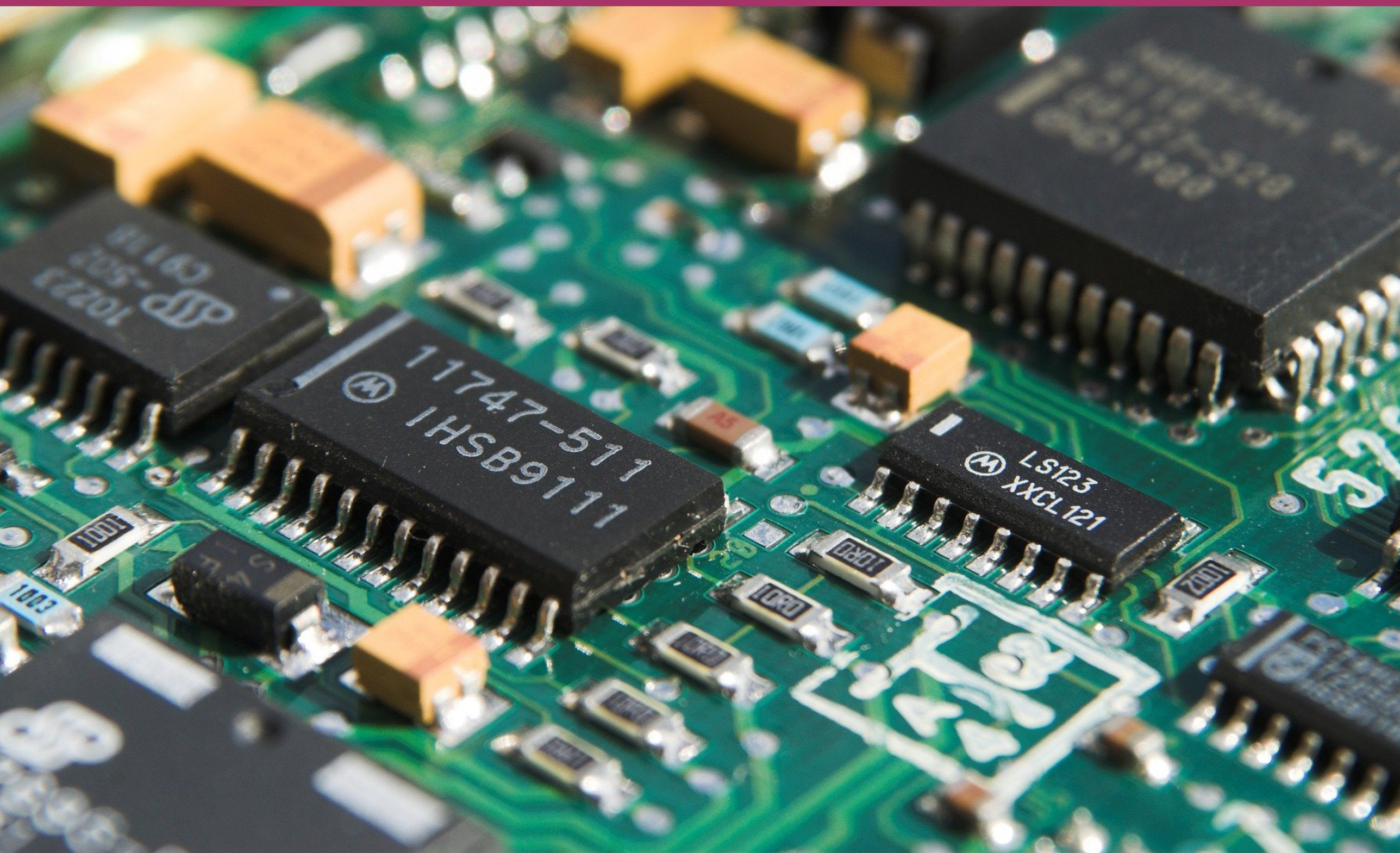
GCSE Computer Science

wjec
cbac



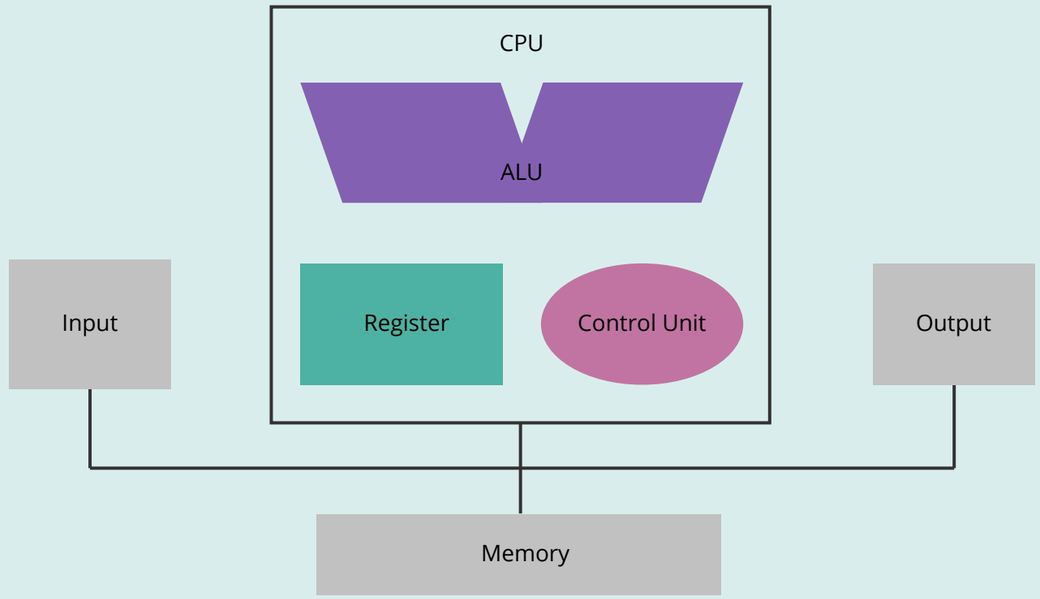
Unit 1: Hardware

Key terminology

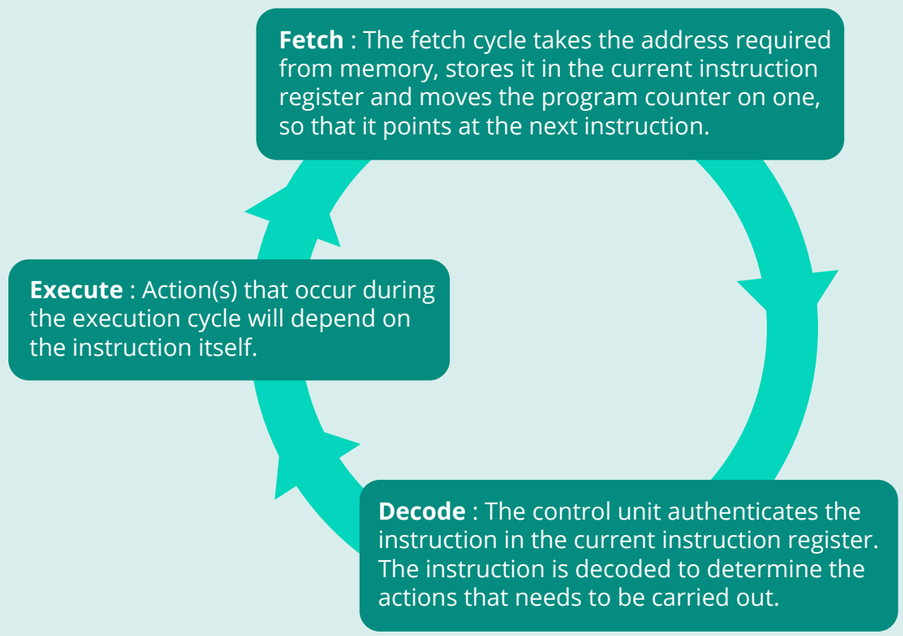
Term	Definition
CPU	The main component in a computer for processing data and instructions.
Control Unit	Directs the flow of instructions and/or data and coordinates the other parts of the CPU. It generates clock ticks or controls the clock.
Arithmetic Logic Unit (ALU)	The ALU performs all the mathematical calculations / logical operations in the CPU.
Registers	Fast access storage locations found on the CPU where data or control information is temporarily stored.
Program Counter (PC)	A counter that keeps track of the memory address of the instruction to be executed next.
Memory Address Register (MAR)	Stores the address in the main memory that is currently being read or written.
Current Instruction Register (CIR)	A temporary holding area for the instruction that has just been fetched from memory.
Memory	Used for the temporary storage of currently running programs and data.
Cache	Incredibly fast, but very expensive volatile memory using in the CPU.
Clock speed	The number of FDE cycles that a CPU can carry out per second.
Cores	Some processors have multiple processors (cores) which can work in parallel, sequentially or can multitask.
Volatile	Stored data is lost when the power is interrupted or switched off.
Permanent	Stored data is kept when the power is interrupted or switched off.

Architecture

The typical Von Neumann architecture



The Fetch-Decode-Execute (FDE) cycle



Performance is affected by greater

- cache size
- clock speed
- number of cores.

Cache size

- Can store more data and instructions.
- It can provide instructions and data to the CPU at a much faster rate (than other system memory such as RAM).

Clock speed

- The FDE cycle will run faster, resulting in more instructions being processed.

Number of cores

- More instructions can be processed at the same time.

NOTE: Performance may be affected where one core is waiting on the result of another and therefore cannot carry out any more instructions.

Unit 1: Hardware

Different types of processor:

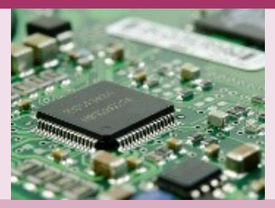
RISC (Reduced instruction Set Computer)	CISC (Complex Instruction Set Computer)
• Carries out more complex commands, but the problem is broken down into simpler instructions.	• Can process a large number of complex instructions. This allows the processor to understand and carry out complex tasks with only a few instructions.
Advantages	
• A RISC processor is able to process these simpler instructions quickly. • Processing simpler instructions also requires less circuitry to decode and execute these instructions, which in turn means less power consumption and therefore less heat being generated.	• A CISC processor is able to process complex instructions, without having to break them down into many simpler instructions. • Processing complex instructions however requires more circuitry to decode and execute these instructions, which in turn means more power consumption and therefore more heat being generated.

Input/output

Input devices	Output devices
Input devices send data to the computer system using devices such as the following:	Output devices receive data from the computer system using devices such as the following:
• Keyboard • Mouse • Scanner • Graphic tablet • Microphone • Touch screen • Webcam	• Speakers • Headphones • Printers • Projectors • Plotters • Video cards • Sound cards

Primary storage

Summary of the different types of memory:

Type	Cache memory	Read-only Memory (ROM)	Random Access Memory (RAM)	Flash memory
				
Volatile or permanent	Volatile	Permanent	Volatile	Permanent
Data can be changed	✓		✓	✓
Relative speed	★★★★	★★★	★★	★
Example use	The temporary storage of frequently accessed data and instructions.	Storing programs such as the system BIOS.	Storing currently running programs and data.	Storing the programs such as the system BIOS.

Secondary storage

Optical		Laser beams are projected onto a disc and if light is reflected back, then data is read as a 1. If light is not reflected back, data is read as a 0.
Magnetic		Data is stored and read using a read-write head and magnetic platter.
Solid state		The technology is called solid state as it doesn't have any moving parts. Its low power consumption and high speed access is advantageous.

Unit 1: Hardware

Secondary storage (continued)

The functional characteristics of contemporary secondary storage devices.

Media	Suitability	Typical capacity	Durability	Portability	Speed
Flash drive 	Moving relatively small files from work to home	2 GB – 512 GB	★★★★	✓	★★★★
External hard drive 	Backing up a home computer system	320 MB – 8 TB	★	✓	★★★
CD/DVD/ Blu-ray disc 	Storing multimedia files	650 MB (CD) 9 GB (DVD) 50 GB (Blu-ray)	★★★	✓	★★
Magnetic tape 	Backing large commercial servers on multiple tapes	200 GB – 400 GB	★★	✓	★

Data storage units

Unit	Symbol	Value
Bit	b	1 bit
Nybble	-	4 bits
Byte	B	8 bits
kilobyte	kB	1024 bytes
Megabyte	MB	1024 kB

Unit	Symbol	Value
Terabyte	TB	1024 GB
Petabyte	PB	1024 TB
Exabyte	EB	1024 PB
Zettabyte	ZB	1024 EB
Yottabyte	YB	1024 ZB

Other hardware components

Graphics Processing Unit (GPU)

Integrated GPU	Dedicated GPU
• Uses the computer's RAM	• Has its own video memory
• Cheaper than installing a dedicated GPU	• Provides the best visual experience
• Generates less heat and uses less power	• Used by people such as professional graphic designers and serious gamers
• Perfect for general graphics processing such as watching or editing videos and word processing.	• Uses more power and requires a good cooling system.

Sound cards

The sound card will convert analogue input signals into digital data and reverse this process for output.

Motherboards

The motherboard is the main circuit board of the computer.

Embedded systems

An embedded system is a combination of software and hardware that performs a specific task whereas a general-purpose computer is designed to carry out multiple tasks.

Examples include

MP3 players, mobile phones, video game consoles, digital cameras, DVD players, and GPS. Household appliances, such as microwave ovens, washing machines and dishwashers.

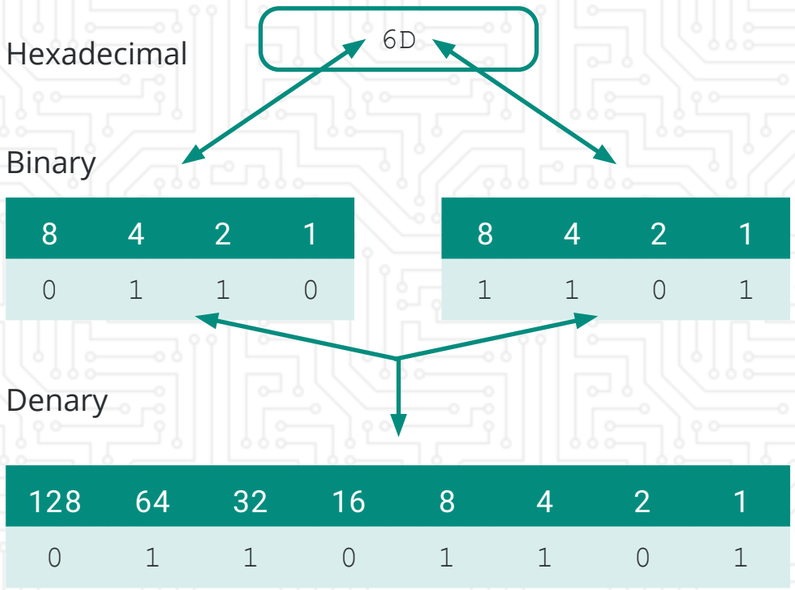
Unit 1: Organisation and structure of data

Key terminology

Term	Definition
Denary	Base 10 number system. Uses digits 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9
Binary	Base 2 number system. Uses digits 0 and 1 only
Hexadecimal	Base 16 number system. Uses digits 0 – 9 and characters A(10), B(11), C(12), D(13), E(14) and F(15) The notation is used as shorthand for binary numbers to avoid errors

Representation of numbers

Conversion between denary, binary and hexadecimal:

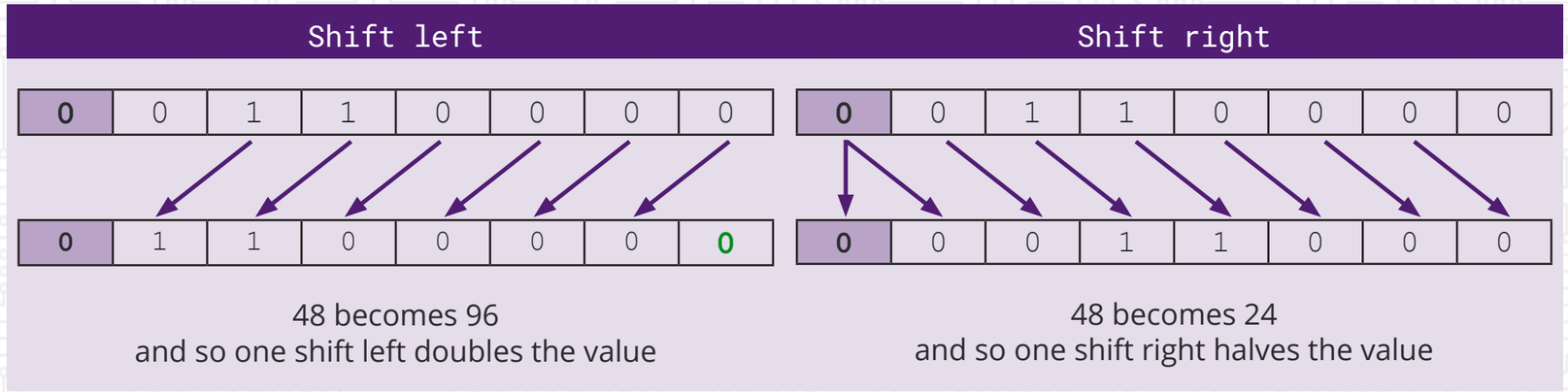


64 + 32 + 8 + 4 + 1 = 109

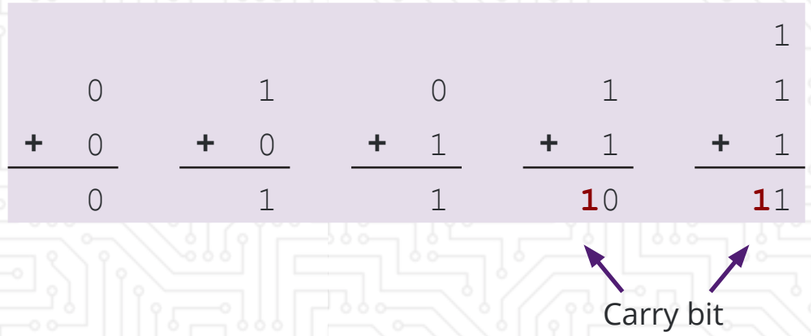
$6D_{16} = 0110110_{12} = 109_{10}$

Arithmetic shift functions

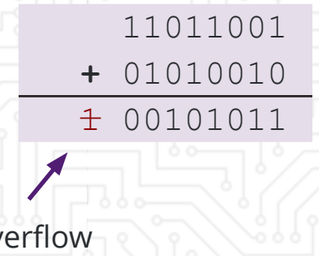
Moving the bits to either the left of the right, doubles (×2) or halves (÷2) the value with each.



Binary addition



Overflow



When the number is too large to be stored by the register, the left most bit cannot be stored and therefore cannot be counted.

Unit 1: Organisation and structure of data

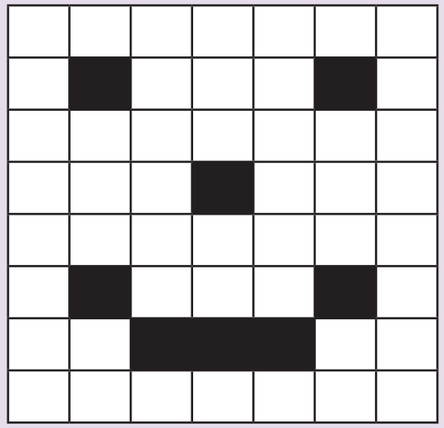
Key terminology

Term	Definition
Pixel	A small coloured dot on a computer display (short for picture elements).
Bitmap	Images are stored as an array of pixels.
Vector	Images that do not store the data by pixels, but are a set of instructions for drawing a geometric shape.
Sample rate	The number of audio samples captured every second.
Bit depth	The number of bits available for each clip.
Bit rate	The number of bits used per second of audio.
Metadata	A set of data that describes and gives information about other data.

Representation of graphics and sound

Digital storage of graphics

A black and white bitmap image will store a 1 for a black pixel and 0 for a white pixel.



0000000
0100010
0000000
0001000
0000000
0100010
0011100
0000000

This bitmap image can be represented using 56 bits (or 7 bytes).

Digital storage of graphics (continued)

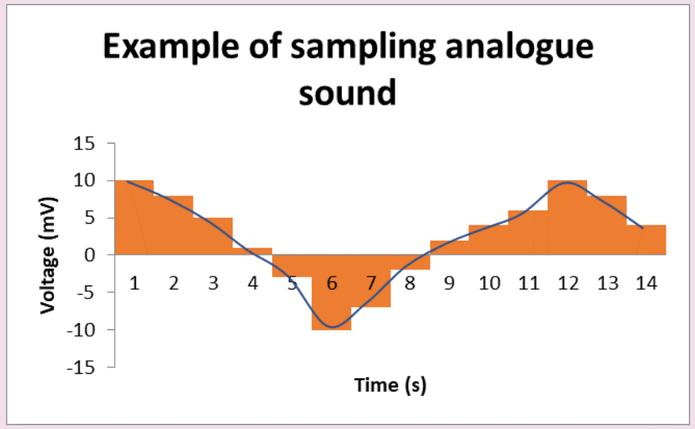
A colour bitmap image is stored using a longer binary number that represents how much red, green and blue (RGB) is required in the colour of each pixel to produce different colours.

The more bits in the binary number, the greater the colour depth, which leads to more colours being available.

Digital storage and sampling of sound

Colour depth	Number of available colours
1 bit	2
2 bits	4
3 bits	8
8 bits	256
16 bits	65,536
24 bits	16.7 million
32 bits	4.3 billion

Sound is converted into a digital signal by a process called sampling. This is where hardware, such as a microphone, measures the level of sound many times per second and records this as binary digits.



The higher the sampling rate, the better the quality, but larger the file size.

Meta data is a set of data that gives information about other data. Examples of metadata in files

- Genre – the genre that the sound file belongs to
- Date Created / Year – the date the graphic was taken
- Location – the location where the graphic was taken
- Size – the original size of the file
- Dimensions.

Unit 1: Organisation and structure of data

Key terminology

Term	Definition
Character	A letter, digit, space, punctuation mark or various other keyboard symbols.
Character set	A table that maps a character with a unique binary number.
Data structure	A specific way of organising data within memory so it can be processed efficiently.
Record	A data structure of related data of different data types.
Primary key	A unique identifier for each record.
Array	A data structure that can hold a fixed number of data items, which must be of the same data type.

Storage of characters

Characters are stored on a computer system as a binary number using a character set. Examples of character sets include ASCII and Unicode.

A small part of the ASCII character set:

Denary	Binary	Hex	Character
64	1000000	40	@
65	1000001	41	A
66	1000010	42	B
67	1000011	43	C

Character sets allow for meaningful data to be exchanged between different computer systems.

Data types

Data type	Description	Examples
Integer	Whole numbers, positive or negative.	42, -11, 0
Real	Numbers, including fractions or decimal points.	12.9, -17.50, 28.0
Boolean	True or false.	1 or 0 TRUE or FALSE
Character	Letter, digit, space, punctuation mark or various other symbols.	'A', 'b', '7','?'
String	A sequence of characters.	'Computer science'

Data structures

Records example

ID*	First name	Surname	Gender	Date of birth
1074	Sara	Davies	F	12/07/2004
1080	Mike	Thomas	M	31/07/1962
1093	Susan	Jones	F	16/08/1958
1123	Dianna	Glanville	F	19/07/1950
1237	Ahmed	Mushtaq	M	21/02/1973

- ID is the primary key field
- Four fields
- Five records
- Different data types.

One-dimensional array example

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
37	11	42	6	26	56	4	76

- 8 elements
- The index always starts at position `[0]`
- Each element can be accessed using its index
- The element at index `[4]` is 26.

Traversing

Print the contents of the array above:

```
1 for i = 0 to 7
2       output myArray[i]
3 next i
```

Insertion

Add data to an element at a given index:

```
1 myArray[4] = 67
```

This would store the value 67 at index [4] of the array.

Deletion

Deleting data from an element at a given index:

```
1 myArray[6] = “”
```

This would leave the memory at index 6 blank.

Searching – arrays can be searched using the index or the value stored at the index.

Unit 1: Organisation and structure of data

Key terminology

Term	Definition
Validation	Ensures that data entered is reasonable.
Verification	Ensures that data entered is consistent.

Two-dimensional array example

	[0]	[1]	[2]	3]	[4]	[5]	[6]	[7]
[0]	37	11	42	6	26	56	4	76
[1]	98	203	64	23	126	79	14	23
[2]	30	1	4	13	29	48	21	211
[3]	10	57	73	110	82	29	289	245

- 32 elements
- Elements are indexed by two numbers, one for its row and one for its column `[y,x]`.
- Each element can be accessed using its index
- The element at index `[1,7]` is 23.

Data validation

Presence checks

Used to check if a required field is left blank.

```
1 if data Entered = “” then
2      output error message
3 end if
```

Format checks

Used to ensure data matches a specific pattern, such as which could be used to validate the format of the postcode.

```
1 if postcode <> format(LL00 0LL) then
2      output error message
3 end if
```

Length checks

Used to ensure an input data string is a sensible length, such as the number of digits in a phone number should be 11.

```
1 if len(telNo) <> 11 then
2      output error message
3 end if
```

Type checks

Used to ensure input data is a particular data type, e.g. quantity ordered to be integer or cost to be real.

Range checks

Used to ensure input data lies within a specified range, such as overtime hours to be > 0 and < 15.

```
1 if hours < 0 OR hours > 15 then
2      output error message
3 end if
```

Data verification

Double-entry

Double entry involves comparing two versions of data input, such as “re-enter your password”.

```
1 if password <> reTypePassword then
2      output error message
3 end if
```

A verification algorithm will compare the two versions and inform the user if they are not identical.

Screen based / visual check

Requires the user to check a display of input data and confirm that it is correct.

Check digit

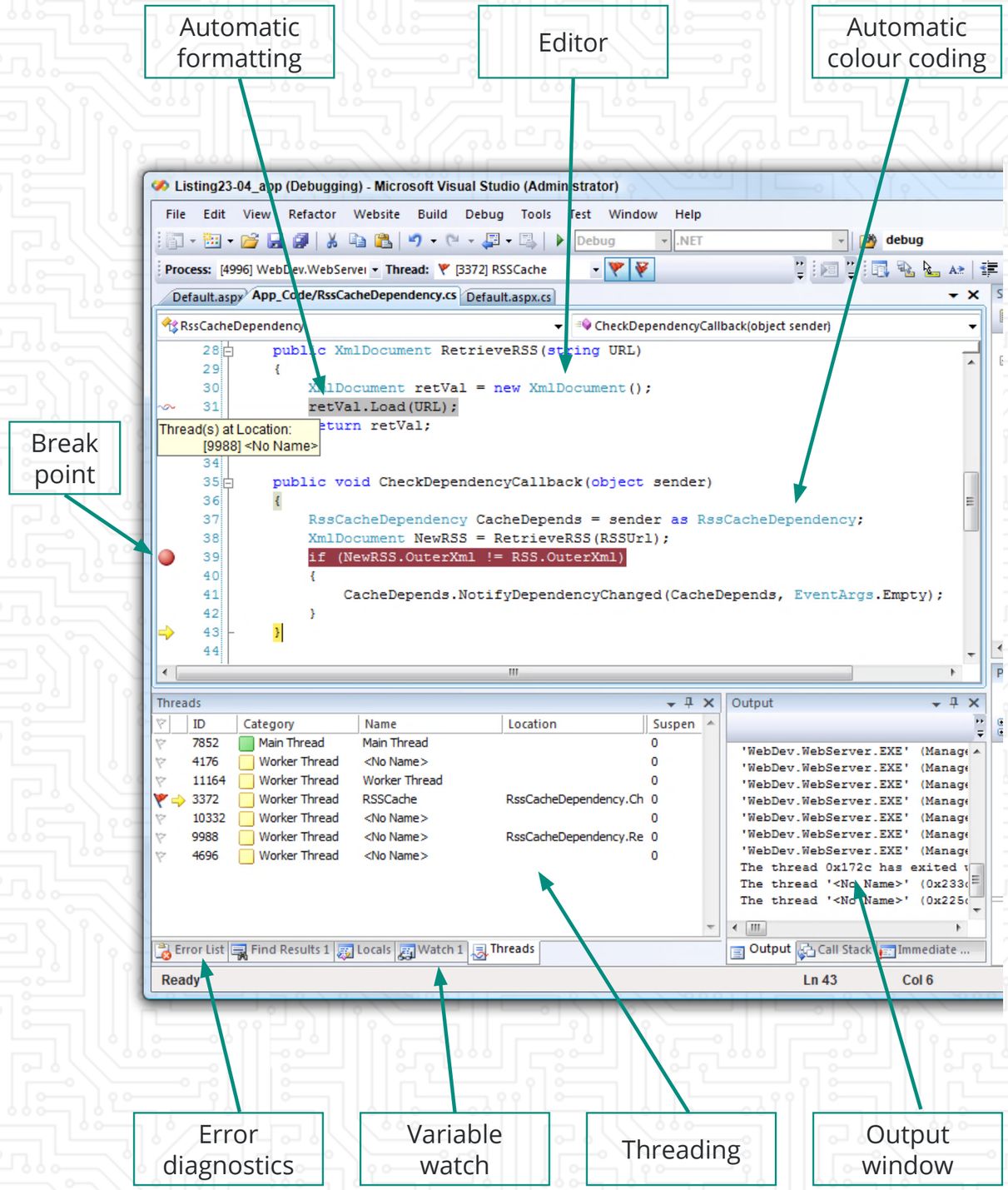
More sophisticated verification algorithms apply calculations to input data, e.g. to produce the check digits of bar codes. Repeating the calculations and checking the result is the same can verify the data.

Unit 1: Software engineering

Key terms

Term	Definition
Editor	Allows a programmer to enter, format and edit source code.
Compiler	Converts source code into executable machine code. Once compiled, a program can be run at any time.
Interpreter	Converts each line of source code into machine code, and executes it as each line of code is run. The conversion process is performed each time the program is run.
Linker	A program which allows previously compiled code, from software libraries, to be linked together.
Loader	A program which loads previously compiled code into memory.
Debugger	A program which helps locate, identify and rectify errors in a program.
Trace	A facility which displays the order in which the lines of a program are executed, and possibly the values of variables as the program is being run.
Break point	Interrupts a program on a specific line of code, allowing the programmer to compare the values of variables against expected values.
Variable watch	A facility that displays the current value of any variable. The value can be 'watched' as the program code is executed line-by-line to see the effects of the code on the variable.
Memory inspector	A facility which will display the contents of a section of memory.
Error diagnostics	Used when a program fails to compile or to run. Error messages are displayed to help the programmer diagnose what has gone wrong.

Example of an integrated development environment (IDE)



Unit 1: Ethical, legal and environmental impacts of digital technology

Key terms

Term	Definition
Ethical	Relating to beliefs about right and wrong and conforming to standards of conduct.
Cybersecurity	Protection against the criminal or unauthorised use of electronic data.
Code of ethics / conduct	Defines acceptable behaviour within an organisation.
Environment	The surroundings or conditions in which a person, animal, or plant lives or operates.

Ethical issues

Digital technology increasingly requires us to consider the ethical issues surrounding its use. The following are examples of this:

- Drones (privacy)
- Self-driving cars (making decisions between life or death for its driver and other people)
- Artificial intelligence (could the creation of thinking machines raise a host of ethical issues including the potential to harm humans).

Code of conduct

It is important for employees to conform to professional standards, including formal and informal codes of ethical behaviour.

Formal codes of ethics are usually enforced by the threat of disciplinary action should the code not be adhered to. Each code of ethics is different and usually reflects an organisation's ethos, values and business style.

Legal issues

Legislation	Overview
The Computer Misuse Act (CMA) 1990	Helps combat issues arising from the misuse of computer systems. The Act makes it an offence to: • access data without permission, e.g. looking at someone else's files • access computer systems without permission, e.g. hacking • alter data stored on a computer system without permission, e.g. writing a virus that deliberately deletes data.
The Freedom of Information Act 2000	People have a right to know about the activities of public authorities, unless there is a good reason for them not to have this information. The Act provides public access to information held by: • Public authorities, who are obliged to publish certain information about their activities.
The Regulation of Investigatory Powers Act (RIPA) 2000	Regulates the powers of public bodies to carry out surveillance and investigation. It also regulates the interception of communications. The Act provides clear legal guidelines for organisations, such as the security services and the police, to carry out surveillance and access the digital communications of individuals, such as email, telephone calls, text messages etc.
The Data Protection Act (DPA) 2018	The DPA applies to all 'personal data'. Personal data is classed as any information relating to a person who can be directly or indirectly identified and so it needs to be protected.

Environmental issues

Example environmental impacts of digital technology on wider society:

- Increase in delivery lorries on the road has caused increased congestion and increases in carbon emissions.
- Are we a paperless society? More and more paper seem to be consumed affecting rainforests and influencing global warming.
- Old computer equipment needs to be disposed of correctly which is expensive. Dumping old computers on landfill sites can cause pollution of toxic substances into the water supply and lead to health problems.
- Computer equipment generates heat so many organisations install air conditioning systems leading to increased carbon emissions.
- Many computers are left on standby, wasting electricity unnecessarily and increasing carbon emissions.
- Mining the rare earth elements required in the manufacture of computers causes pollution.
- Global assembly lines and pollution from transportation.

Unit 2: Programming Languages

Markup languages

HTML is a standard used when creating web pages.

HTML tags commonly come in pairs. The first tag in a pair is called the opening tag and the second tag is called the closing tag.

Opening tag	Closing tag	Description
<html>	</html>	Defines the root of an HTML document
<title>	</title>	Defines a title for the document
<body>	</body>	Defines the document's body
<h1> <h6>	</h1> </h6>	Defines HTML headings h1 to h6
		Defines a paragraph
		Italicises a part of text in an alternate voice or mood
		Defines bold text
		Defines centred text
		Defines a hyperlink and specifies the URL of the page the link goes to
		Defines a hyperlink and specifies the email address the link goes to
		Defines an unordered list
		Defines a list item
<blockquote>	</blockquote>	Defines a section that is quoted from another source
<hr>	N/A	Defines a thematic change in the content
<img src="…">	N/A	Defines an image

Using markup language:

TrackMyPetCare.com

The new online system for keeping track of all your pets' needs, including:

- Vet check-ups
- Booster dates
- Birthdays and other anniversaries!

Click the link below to find out more:

www.TrackMyPetCare.com

finalPet - Notepad

File Edit Format View Help

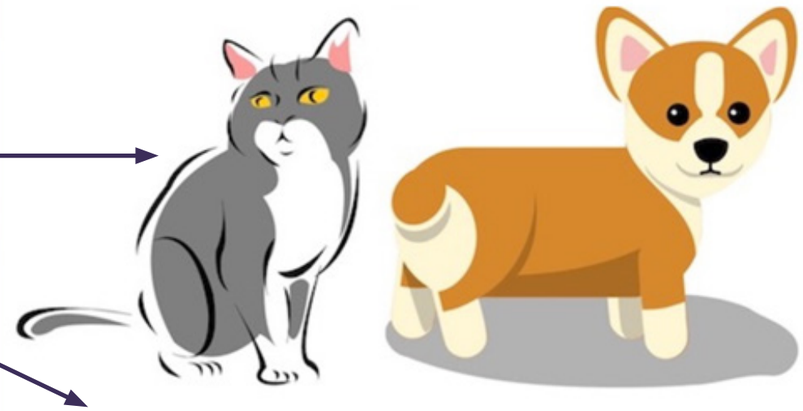
```
<html>
<head>
<title>
Track My Pet Care
</title>
</head>

<body>
<center>
<h1>TrackMyPetCare.com
</h1>
</center>
<img src="pets.jpg">
<p>
The new online system for keeping track of all
your pets' needs. Including:
</p>
<p>
<ul>
<li> Vet check-ups</li>
<li> Booster dates</li>
<li> Birthday and other Anniversaries!</li>
</ul>
</p>
<p>
Click the link below to find out more:
</p>
<p> <a href="http://www.TrackMyPetCare.com">
www.TrackMyPetCare.com
</a></p>
</body>
</html>
```

Track My Pet Care

File

Apps

TrackMyPetCare.com



The new online system for keeping track of all your pets' needs. Including:

- Vet check-ups
- Booster dates
- Birthday and other Anniversaries!

Click the link below to find out more:

www.TrackMyPetCare.com

Unit 2: Programming Languages

Assembly language

Assembly language is a programming language that is once removed from machine code. Machine code is made up of 0s and 1s and is extremely difficult for a programmer to use. Assembly language has the same structure and instruction set as the commands in machine code but they use mnemonics (names) rather than binary code.

Mnemonic	Description
INP	Inputs a value and stores it in the accumulator
OUT	Displays the contents of the accumulator
STA	Transfers a number from the accumulator to RAM
LDA	Transfers a number from RAM to the accumulator
ADD	Adds the contents of the accumulator to the contents of a RAM address
SUB	Subtracts the contents of the accumulator from the contents of a RAM address
BRA	Jumps to the RAM location specified – used for loops
HLT	Stops the processor
DAT	Defines variables

Benefits of using assembly language:

- require less memory and execution time
- allow code to interact directly with hardware, such as device drivers
- suitable for time-critical processes.

Using assembly language:

This is an assembly language program to add together two numbers.

INP	Input a number
STA 1A	Store the number in memory location 1A
INP	Input a number
ADD 1A	Add this number to the number stored in memory location 1A
OUT	Output the result
HLT	Stop

This is an assembly language program for the subtraction of one number from another.

INP	input a number
STA first	store the number in a variable called ‘first’
INP	input a number
STA second	store the number in a variable called ‘second’
LDA first	load the number in the ‘first’ variable into the accumulator
SUB second	subtract the contents of the ‘second’ variable from the accumulator
OUT	output the number in the accumulator
first DAT	declare ‘first’ as a variable
Second DAT	declare ‘second’ as a variable